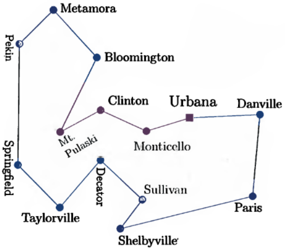
Arora's PTAS for the Euclidean TSP[1]

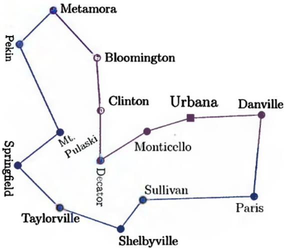
R. Inkulu
http://www.iitg.ac.in/rinkulu/

[1]most of the figures in this presentation are from the references mentioned

problem description

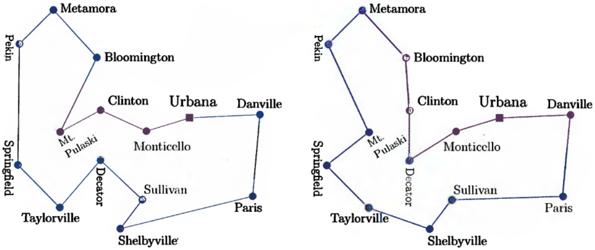
Given a set P of points in Euclidean plane, find a tour of minimum cost that visits all the points of P.





problem description

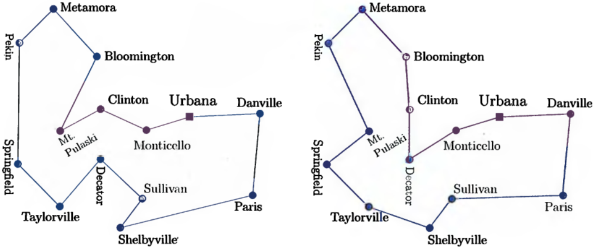
Given a set P of points in Euclidean plane, find a tour of minimum cost that visits all the points of P.



- known to be NP-hard
- objective of this talk: algo (PTAS) to compute an *approximate tour* whose *expected length* is $(1+\epsilon)\tau_{OPT}$ in $O(n(\frac{\lg n}{\epsilon})^{O(\frac{1}{\epsilon})})$ time

problem description

Given a set P of points in Euclidean plane, find a tour of minimum cost that visits all the points of P.



- known to be NP-hard
- objective of this talk: algo (PTAS) to compute an *approximate tour* whose *expected length* is $(1+\epsilon)\tau_{OPT}$ in $O(n(\frac{\lg n}{\epsilon})^{O(\frac{1}{\epsilon})})$ time

for the convenience, we let the following:

$1 > \epsilon > \frac{1}{n}$

P is contained in $[\frac{1}{2}, \frac{1}{2}] \times [1, 1]$

$diam(P) \geq \frac{1}{4}$

Outline

1. algorithm

2. analysis

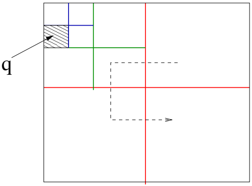
round the input points to grid points

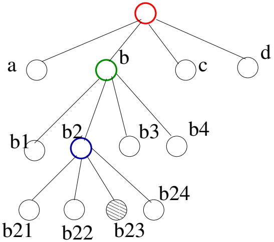
- set up a grid with sidelength $G = \{\frac{1}{n\lceil\frac{32}{\epsilon}\rceil}(i,j) | i,j \text{ are integers}\}$

round the input points to grid points

- set up a grid with sidelength $G = \{\frac{1}{n\lceil\frac{32}{\epsilon}\rceil}(i,j) | i,j \text{ are integers}\}$
- round (snap) each point in P to an arbitrary corner (grid point) of the grid cell in which it is located $\rightarrow$ the resultant point set is Q
  - this would guarantee the quadtree to be constructed for Q to have logarithmic depth

setting up a quadtree over Q

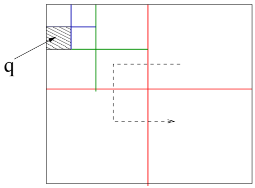


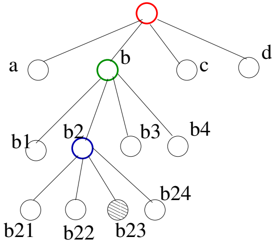


- consider the unit square with its southwest corner at $(\frac{1}{2}, \frac{1}{2})^2$ as the square of the root node

[2] later, this corner will be fixed at a point randomly chosen from the square with the southwest and northeast corners respectively at $(0, 0)$ and $(\frac{1}{2}, \frac{1}{2})$

setting up a quadtree over Q

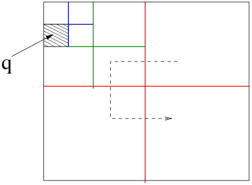


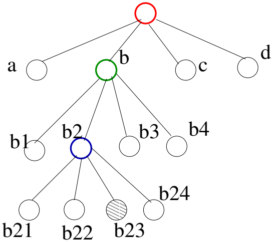


- consider the unit square with its southwest corner at $(\frac{1}{2}, \frac{1}{2})^2$ as the square of the root node
- since the minimal distance between a pair of points of Q is at least $\frac{1}{n\lceil\frac{32}{\epsilon}\rceil}$, it suffices to have the leafnode square sidelength $\frac{1}{n\lceil\frac{32}{\epsilon}\rceil}$

[2] later, this corner will be fixed at a point randomly chosen from the square with the southwest and northeast corners respectively at $(0, 0)$ and $(\frac{1}{2}, \frac{1}{2})$

setting up a quadtree over Q



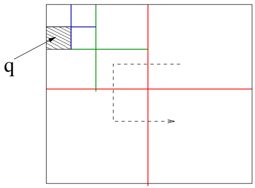


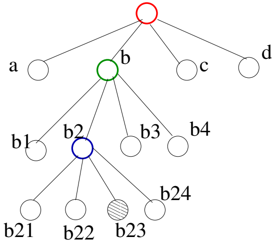
- consider the unit square with its southwest corner at $(\frac{1}{2},\frac{1}{2})$[2] as the square of the root node
- since the minimal distance between a pair of points of Q is at least $\frac{1}{n\lceil\frac{32}{\epsilon}\rceil}$, it suffices to have the leafnode square sidelength $\frac{1}{n\lceil\frac{32}{\epsilon}\rceil}$
- height H of the quadtree is $O(\lg n)$

  level of root is 0; for every $0 \le i \le H$, grid corresponding to level i is denoted with G^i and the length of any edge of G^i is $\frac{1}{2^i}$

[2] later, this corner will be fixed at a point randomly chosen from the square with the southwest and northeast corners respectively at $(0,0)$ and $(\frac{1}{2},\frac{1}{2})$

setting up a quadtree over Q



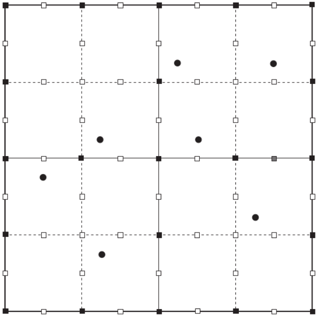


- consider the unit square with its southwest corner at $(\frac{1}{2}, \frac{1}{2})$[2] as the square of the root node
- since the minimal distance between a pair of points of Q is at least $\frac{1}{n\lceil\frac{32}{\epsilon}\rceil}$, it suffices to have the leafnode square sidelength $\frac{1}{n\lceil\frac{32}{\epsilon}\rceil}$
- height H of the quadtree is $O(\lg n)$

  level of root is 0; for every $0 \leq i \leq H$, grid corresponding to level i is denoted with G^i and the length of any edge of G^i is $\frac{1}{2^i}$
- letting no node of the quadtree is empty, the number of nodes is $O(n \lg n)$

[2] later, this corner will be fixed at a point randomly chosen from the square with the southwest and northeast corners respectively at $(0, 0)$ and $(\frac{1}{2}, \frac{1}{2})$

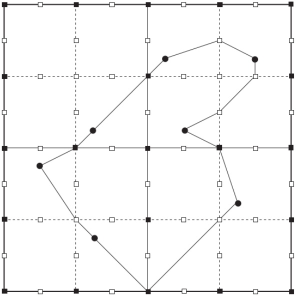
portals along the quadtree square sides



introduce equi-spaced $m = \frac{20H}{\epsilon}$ (which is $O(\frac{\lg n}{\epsilon})$) *portals* along each side of each quadtree square together with four at the corners of the same [3]

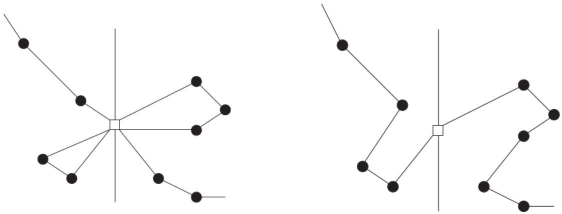
[3] choosing $m+1$ as a power of two, each portal on the sides of a level $i-1$ square are at the same location as a portal on the side of some level i square contained in the level $i-1$ square

requisite characteristics of a constrained tour



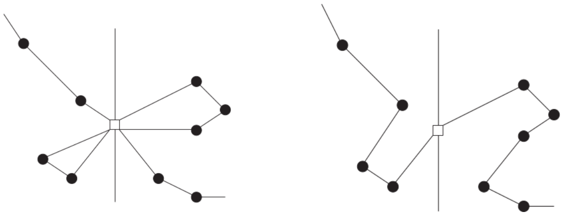
- tour is permitted to cross an edge e of a quadree square only via portals on e

requisite characteristics of a constrained tour (cont)



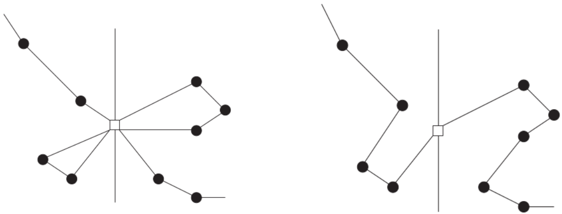
- no portal can be used more than twice

requisite characteristics of a constrained tour (cont)



- no portal can be used more than twice
- tour can use at most $r = \frac{90}{\epsilon}$ i.e., $O(\frac{1}{\epsilon})$ portals corresponding to any side of any quadtree square i.e., tour need to be *r-light* w.r.t. any quadtree square edge

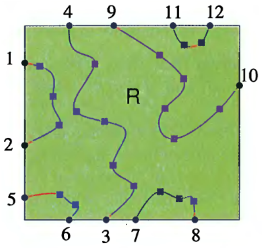
requisite characteristics of a constrained tour (cont)



- no portal can be used more than twice
- tour can use at most $r = \frac{90}{\epsilon}$ i.e., $O(\frac{1}{\epsilon})$ portals corresponding to any side of any quadtree square i.e., tour need to be *r-light* w.r.t. any quadtree square edge

  is it possible to have such a tour while being a good apprx to an optimal tour?

nested parenthesis structure of portal-to-portal paths

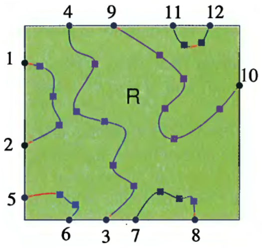


$$(4(1,2)(5,6)3)(7,8)(9(11,12)10)$$

$$(_4(_1)_2(_5)_6)_3(_7)_8(_9(_{11})_{12})_{10}$$

- portal-to-portal paths follow parentization; hence, one can try all settings of parenthesis, translate these into possible layouts of paths and discard the ones that have intersecting paths

nested parenthesis structure of portal-to-portal paths



$$(4(1,2)(5,6)3)(7,8)(9(11,12)10)$$

$$(_4(_1)_2(_5)_6)_3(_7)_8(_9(_{11})_{12})_{10}$$

- portal-to-portal paths follow parentization; hence, one can try all settings of parenthesis, translate these into possible layouts of paths and discard the ones that have intersecting paths

- therefore, the number of layouts of k portal-portal paths in grid square is the k^{th} Catalan number, which is $\frac{1}{k+1}\binom{2k}{k} = O(2^{2k})$

valid solutions of any grid square

- for every 0 to $4r$ portal combination of $4m+4$ portals on the boundary, considering that each chosen portal can be used 1 to 2 times, for every permutation p of the chosen portals, alternately mark portals in p with enter-exit $\rightarrow$ number of subproblems $\leq \sum_{i=0}^{8r} \binom{2(4m+4)}{i} i!$ i.e., $\left(\frac{1}{\epsilon} \lg n\right)^{O\left(\frac{1}{\epsilon}\right)}$

[4] a collection of subpaths (which intersect this square) of the apprx tour being constructed

valid solutions of any grid square

- for every 0 to $4r$ portal combination of $4m+4$ portals on the boundary, considering that each chosen portal can be used 1 to 2 times, for every permutation p of the chosen portals, alternately mark portals in p with enter-exit $\rightarrow$ number of subproblems $\leq \sum_{i=0}^{8r} \binom{2(4m+4)}{i} i!$ i.e., $(\frac{1}{\epsilon} \lg n)^{O(\frac{1}{\epsilon})}$

- find a minimum cost feasible subtour[4]
  - feasibility criteria: precisely satisfies the portal marking of that quadtree square; non-intersecting; subtours cover points belonging to quadtree square

[4] a collection of subpaths (which intersect this square) of the apprx tour being constructed

valid solutions of any grid square

- for every 0 to $4r$ portal combination of $4m+4$ portals on the boundary, considering that each chosen portal can be used 1 to 2 times, for every permutation p of the chosen portals, alternately mark portals in p with enter-exit $\rightarrow$ number of subproblems $\leq \sum_{i=0}^{8r} \binom{2(4m+4)}{i} i!$ i.e., $\left(\frac{1}{\epsilon} \lg n\right)^{O\left(\frac{1}{\epsilon}\right)}$

- find a minimum cost feasible subtour[4]
  - feasibility criteria: precisely satisfies the portal marking of that quadtree square; non-intersecting; subtours cover points belonging to quadtree square

— to achieve efficiency, build the subtours in bottom-to-top fashion: using memoization of DP, from leaf nodes to root of the quadtree

[4] a collection of subpaths (which intersect this square) of the apprx tour being constructed

dynamic programming over squares of quadtree grid

- interaction between problems is according to the organization of quadtree nodes

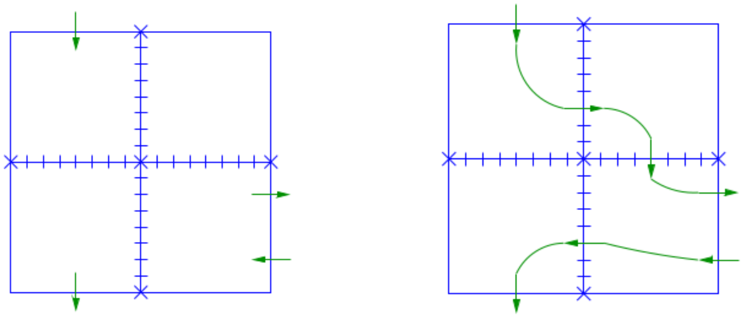
dynamic programming over squares of quadtree grid

- interaction between problems is according to the organization of quadtree nodes
- memoize the subproblem solutions

dynamic programming over squares of quadtree grid

- interaction between problems is according to the organization of quadtree nodes
- memoize the subproblem solutions
- base case: if a square has $O(\frac{1}{\epsilon})$ points of Q, then solve that subprobelm by brute force

DP: assemble subtours from children



- at every non-leaf node, for every chosen portal permutation, save these entries in the DP table: portal permutations of each child node which together caused the minimum cost subtour at the parent

time complexity

- number of subproblems to pursue at each quadtree node $\leq \sum_{i=0}^{8r} \binom{2(4m+4)}{i} i!$ i.e., $(\frac{1}{\epsilon} \lg n)^{O(\frac{1}{\epsilon})})$

time complexity

- number of subproblems to pursue at each quadtree node
$\leq \sum_{i=0}^{8r} \binom{2(4m+4)}{i} i!$ i.e., $(\frac{1}{\epsilon} \lg n)^{O(\frac{1}{\epsilon})}$
- subproblem at each quadtree node (ignoring the recursive subproblems) can be solved in $((\frac{1}{\epsilon} \lg n)^{O(\frac{1}{\epsilon})})^{O(1)}$ time
  - assembling: each subproblem solution of 1^{st} child with each ... with each subproblem solution of 4^{th} child
  - and, checking whether the assembled subtour is according to enter-exit constraints and non-intersecting within the quadtree square

time complexity

- number of subproblems to pursue at each quadtree node $\le \sum_{i=0}^{8r} \binom{2(4m+4)}{i} i!$ i.e., $(\frac{1}{\epsilon} \lg n)^{O(\frac{1}{\epsilon})}$
- subproblem at each quadtree node (ignoring the recursive subproblems) can be solved in $((\frac{1}{\epsilon} \lg n)^{O(\frac{1}{\epsilon})})^{O(1)}$ time
  - assembling: each subproblem solution of 1^{st} child with each ... with each subproblem solution of 4^{th} child
  - and, checking whether the assembled subtour is according to enter-exit constraints and non-intersecting within the quadtree square
- since there are $O(n \lg n)$ nodes in the quadtree, the time is $n(\frac{1}{\epsilon} \lg n)^{O(\frac{1}{\epsilon})}$ (including the time to brute-force enumeration at leaf nodes)

Outline

1. algorithm

2. analysis

factors causing errors

(i) building a tour π_{opt}^{Q} of points of Q instead of a tour π_{opt} of points of P

factors causing errors

(i) building a tour π^{Q}_{opt} of points of Q instead of a tour π_{opt} of points of P

(ii) limiting the tour to be r-light

factors causing errors

(i) building a tour π_{opt}^{Q} of points of Q instead of a tour π_{opt} of points of P

(ii) limiting the tour to be r-light

(iii) limiting tour intersection with the edges of the quadtree squares to portals

factors causing errors

(i) building a tour π_{opt}^{Q} of points of Q instead of a tour π_{opt} of points of P

(ii) limiting the tour to be r-light

(iii) limiting tour intersection with the edges of the quadtree squares to portals

these are upper bounded by transforming π_{opt} while making it to obey these specific constraints

factors causing errors

(i) building a tour π^Q_{opt} of points of Q instead of a tour π_{opt} of points of P

(ii) limiting the tour to be r-light

(iii) limiting tour intersection with the edges of the quadtree squares to portals

these are upper bounded by transforming π_{opt} while making it to obey these specific constraints

We denote $\|\pi_{opt}\|$ with OPT and $\|\pi^Q_{opt}\|$ with OPT_Q.

error type (i): rounding points in P to get to Q

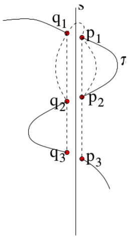
- The error in making π_{opt}^{Q} to touch points of P is

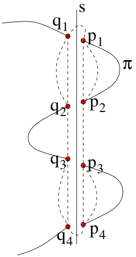
$$\leq n(2\frac{\sqrt{2}}{n\lceil\frac{32}{\epsilon}\rceil})$$

$$\leq \frac{4\epsilon}{32}$$

$$\leq \frac{\epsilon}{2}OPT \quad \text{(since } OPT \geq diam(P) \text{ which is } \tfrac{1}{4}\text{)}$$

patching lemma: patch the path with respect to a line segment

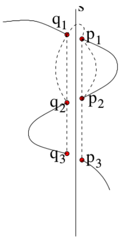


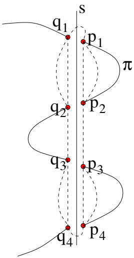


find an Eulerian tour in the graph

- replace a polyline π that crosses a line segment s at least three times with a polyline π' that crosses s at most twice: construct an Eulerian graph by introducing a few edges (see the above Fig.) to π
  $\|\pi'\| \leq \|\pi\| + (\sum_{i \text{ being odd}} \|p_i p_{i+1}\| + \|q_i q_{i+1}\|) + 2\|s\| \leq \|\pi\| + 4\|s\|$
  __________ (0)

patching lemma: patch the path with respect to a line segment

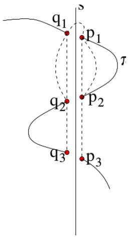


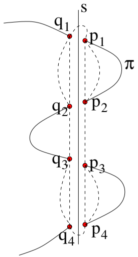


find an Eulerian tour in the graph

- replace a polyline π that crosses a line segment s at least three times with a polyline π' that crosses s at most twice: construct an Eulerian graph by introducing a few edges (see the above Fig.) to π
$\|\pi'\| \leq \|\pi\| + (\sum_{i \text{ being odd}} \|p_i p_{i+1}\| + \|q_i q_{i+1}\|) + 2\|s\| \leq \|\pi\| + 4\|s\|$
———— (0)
- corollary: as the length of any edge of G^i is $\frac{1}{2^i}$, error in patching any one edge of G^i is $\leq \frac{4}{2^i}$

patching lemma: patch the path with respect to a line segment





find an Eulerian tour in the graph

- replace a polyline π that crosses a line segment s at least three times with a polyline π' that crosses s at most twice: construct an Eulerian graph by introducing a few edges (see the above Fig.) to π
  $\|\pi'\| \leq \|\pi\| + (\sum_{i \text{ being odd}} \|p_i p_{i+1}\| + \|q_i q_{i+1}\|) + 2\|s\| \leq \|\pi\| + 4\|s\|$ ———— (0)
- corollary: as the length of any edge of G^i is $\frac{1}{2^i}$, error in patching any one edge of G^i is $\leq \frac{4}{2^i}$
- corollary: viewing the portal is of zero lengthed segment, any optimal solution need to use a portal at most twice (hence, we are allowing 2 intersections per portal in the apprx tour)

patching grid edges at every level of the quadtree

- while considering quadtree nodes in bottom-to-top[5], at every level $H \geq i \geq 1$[6], patch the tour π_{opt} w.r.t. every grid edge of G^i:

  if the current tour intersects e more than r times so that it intersects e at most twice after patching

[5] when we fix an edge of a grid so that the tour does not intersect it too many times, the number of times the patched tour crosses boundaries of higher-level nodes of the quadtree also goes down; similarly, the total number of crossings (of the tour with the grids) drop exponentially as we use larger and larger grids; thus requiring fewer fix-ups; thus, intuitively, one can think about all the patching happening in the bottom level of the quadtree

[6] G^0 does not intersect the generated path

patching grid edges at every level of the quadtree

- while considering quadtree nodes in bottom-to-top[5], at every level $H \geq i \geq 1$[6], patch the tour π_{opt} w.r.t. every grid edge of G^i:

  if the current tour intersects e more than r times so that it intersects e at most twice after patching

  let π_i denote the resulting path after patching at G^i

  let π_{i+1} denote the path just before patching at G^i

  π_{H+1} is π_{opt}, and π_1 is the patched tour after patching at every level from H to 1

[5] when we fix an edge of a grid so that the tour does not intersect it too many times, the number of times the patched tour crosses boundaries of higher-level nodes of the quadtree also goes down; similarly, the total number of crossings (of the tour with the grids) drop exponentially as we use larger and larger grids; thus requiring fewer fix-ups; thus, intuitively, one can think about all the patching happening in the bottom level of the quadtree

[6] G^0 does not intersect the generated path

exploiting sliding grid

randomly choose the unit grid origin from $[0, 0] \times [\frac{1}{2}, \frac{1}{2}]$:

- for every level $i \geq 1$ of the quadtree, the expected number of intersections of a polyline w with the vertical and horizontal edges of G^i is at most $2^{i+1}\|w\|$ $\leftarrow$ probabilistic argument is due to sliding grid

  ———————— (1)

exploiting sliding grid

randomly choose the unit grid origin from $[0, 0] \times [\frac{1}{2}, \frac{1}{2}]$:

- for every level $i \geq 1$ of the quadtree, the expected number of intersections of a polyline w with the vertical and horizontal edges of G^i is at most $2^{i+1}\|w\|$ $\leftarrow$ probabilistic argument is due to sliding grid

  ———————— (1)

- if a point p belongs to open edges of G^i then the point p belongs to open edges of G^{i-1} with probability $\frac{1}{2}$ (as every grid edge of G^i has probability $\frac{1}{2}$ to survive and be a grid edge of G^{i-1})

  ———————— (2)

error type (ii): patching π_{opt} at every level from bottom-to-top

F_i: number of patching operations performed at i^{th} level; Y_i: number of times the current tour crosses the edges of G^i just before considering G^i for patching; n_i: number of times the current tour crosses the edges of G^i after patching

error type (ii): patching π_{opt} at every level from bottom-to-top

F_i: number of patching operations performed at i^{th} level; Y_i: number of times the current tour crosses the edges of G^i just before considering G^i for patching; n_i: number of times the current tour crosses the edges of G^i after patching

expected increase in length of π_{opt} due to patchings at every level

error type (ii): patching π_{opt} at every level from bottom-to-top

F_i: number of patching operations performed at i^{th} level; Y_i: number of times the current tour crosses the edges of G^i just before considering G^i for patching; n_i: number of times the current tour crosses the edges of G^i after patching

expected increase in length of π_{opt} due to patchings at every level

$$= E[\sum_{i=1}^{H} \frac{4F_i}{2^i}] \quad \text{(due to (0))}$$

error type (ii): patching π_{opt} at every level from bottom-to-top

F_i: number of patching operations performed at i^{th} level; Y_i: number of times the current tour crosses the edges of G^i just before considering G^i for patching; n_i: number of times the current tour crosses the edges of G^i after patching

expected increase in length of π_{opt} due to patchings at every level

$$= E[\sum_{i=1}^{H} \frac{4F_i}{2^i}] \quad \text{(due to (0))}$$

$$\leq E[2F_1 + \sum_{i=2}^{H} \frac{4F_i}{2^i}]$$

error type (ii): patching π_{opt} at every level from bottom-to-top

F_i: number of patching operations performed at i^{th} level; Y_i: number of times the current tour crosses the edges of G^i just before considering G^i for patching; n_i: number of times the current tour crosses the edges of G^i after patching

expected increase in length of π_{opt} due to patchings at every level

$$= E[\sum_{i=1}^{H} \frac{4F_i}{2^i}] \quad \text{(due to (0))}$$

$$\leq E[2F_1 + \sum_{i=2}^{H} \frac{4F_i}{2^i}]$$

$$\leq E[2F_1] + \frac{4}{r-2} \sum_{i=2}^{H} \frac{E[Y_i]-2E[Y_{i-1}]}{2^i}$$

(noting that due to (2), $E[Y_{i-1}] = E[E[Y_{i-1}|n_i]] = E[\frac{n_i}{2}]$, and $E[Y_i] = E[\frac{n_i}{2}] \leq E[\frac{Y_i-(r-2)F_i}{2}] = \frac{1}{2}E[Y_i] - \frac{k-2}{2}E[F_i]$)

error type (ii): patching π_{opt} at every level from bottom-to-top

F_i: number of patching operations performed at i^{th} level; Y_i: number of times the current tour crosses the edges of G^i just before considering G^i for patching; n_i: number of times the current tour crosses the edges of G^i after patching

expected increase in length of π_{opt} due to patchings at every level

$$= E[\sum_{i=1}^{H} \frac{4F_i}{2^i}] \quad \text{(due to (0))}$$

$$\leq E[2F_1 + \sum_{i=2}^{H} \frac{4F_i}{2^i}]$$

$$\leq E[2F_1] + \frac{4}{r-2} \sum_{i=2}^{H} \frac{E[Y_i]-2E[Y_{i-1}]}{2^i}$$

(noting that due to (2), $E[Y_{i-1}] = E[E[Y_{i-1}|n_i]] = E[\frac{n_i}{2}]$, and $E[Y_i] = E[\frac{n_i}{2}] \leq E[\frac{Y_i-(r-2)F_i}{2}] = \frac{1}{2}E[Y_i] - \frac{k-2}{2}E[F_i]$)

$$\leq E[2\frac{Y_1}{r-2}] + \frac{4}{r-2}\left(\frac{E[Y_H]}{2^H} - \frac{E[Y_1]}{2}\right) \quad \text{(since } F_1 \leq \frac{Y_1}{r} \leq \frac{Y_1}{(r-2)}\text{)}$$

error type (ii): patching π_{opt} at every level from bottom-to-top

F_i: number of patching operations performed at i^{th} level; Y_i: number of times the current tour crosses the edges of G^i just before considering G^i for patching; n_i: number of times the current tour crosses the edges of G^i after patching

expected increase in length of π_{opt} due to patchings at every level

$$= E[\sum_{i=1}^{H} \frac{4F_i}{2^i}] \quad \text{(due to (0))}$$

$$\leq E[2F_1 + \sum_{i=2}^{H} \frac{4F_i}{2^i}]$$

$$\leq E[2F_1] + \frac{4}{r-2} \sum_{i=2}^{H} \frac{E[Y_i]-2E[Y_{i-1}]}{2^i}$$

(noting that due to (2), $E[Y_{i-1}] = E[E[Y_{i-1}|n_i]] = E[\frac{n_i}{2}]$, and

$E[Y_i] = E[\frac{n_i}{2}] \leq E[\frac{Y_i-(r-2)F_i}{2}] = \frac{1}{2}E[Y_i] - \frac{k-2}{2}E[F_i]$)

$$\leq E[2\frac{Y_1}{r-2}] + \frac{4}{r-2}(\frac{E[Y_H]}{2^H} - \frac{E[Y_1]}{2}) \quad \text{(since } F_1 \leq \frac{Y_1}{r} \leq \frac{Y_1}{(r-2)}\text{)}$$

$$= \frac{4}{r-2}(\frac{E[Y_1]}{2} + \frac{E[Y_H]}{2^H} - \frac{E[Y_1]}{2})$$

error type (ii): patching π_{opt} at every level from bottom-to-top

F_i: number of patching operations performed at i^{th} level; Y_i: number of times the current tour crosses the edges of G^i just before considering G^i for patching; n_i: number of times the current tour crosses the edges of G^i after patching

expected increase in length of π_{opt} due to patchings at every level

$$= E[\sum_{i=1}^{H} \frac{4F_i}{2^i}] \quad \text{(due to (0))}$$

$$\leq E[2F_1 + \sum_{i=2}^{H} \frac{4F_i}{2^i}]$$

$$\leq E[2F_1] + \frac{4}{r-2} \sum_{i=2}^{H} \frac{E[Y_i] - 2E[Y_{i-1}]}{2^i}$$

(noting that due to (2), $E[Y_{i-1}] = E[E[Y_{i-1}|n_i]] = E[\frac{n_i}{2}]$, and

$E[Y_i] = E[\frac{n_i}{2}] \leq E[\frac{Y_i - (r-2)F_i}{2}] = \frac{1}{2}E[Y_i] - \frac{k-2}{2}E[F_i]$)

$$\leq E[2\frac{Y_1}{r-2}] + \frac{4}{r-2}\left(\frac{E[Y_H]}{2^H} - \frac{E[Y_1]}{2}\right) \quad \text{(since } F_1 \leq \frac{Y_1}{r} \leq \frac{Y_1}{(r-2)}\text{)}$$

$$= \frac{4}{r-2}\left(\frac{E[Y_1]}{2} + \frac{E[Y_H]}{2^H} - \frac{E[Y_1]}{2}\right)$$

$$\leq \frac{4}{r-2}\frac{E[Y_H]}{2^H} \quad \text{(note that } Y_H \text{ is the number of times } \pi_{opt} \text{ crosses the edges of } G^H\text{)}$$

error type (ii): patching π_{opt} at every level from bottom-to-top

F_i: number of patching operations performed at i^{th} level; Y_i: number of times the current tour crosses the edges of G^i just before considering G^i for patching; n_i: number of times the current tour crosses the edges of G^i after patching

expected increase in length of π_{opt} due to patchings at every level

$$= E[\textstyle\sum_{i=1}^{H} \frac{4F_i}{2^i}] \quad \text{(due to (0))}$$

$$\leq E[2F_1 + \textstyle\sum_{i=2}^{H} \frac{4F_i}{2^i}]$$

$$\leq E[2F_1] + \frac{4}{r-2} \textstyle\sum_{i=2}^{H} \frac{E[Y_i]-2E[Y_{i-1}]}{2^i}$$

(noting that due to (2), $E[Y_{i-1}] = E[E[Y_{i-1}|n_i]] = E[\frac{n_i}{2}]$, and

$E[Y_i] = E[\frac{n_i}{2}] \leq E[\frac{Y_i-(r-2)F_i}{2}] = \frac{1}{2}E[Y_i] - \frac{k-2}{2}E[F_i]$)

$$\leq E[2\frac{Y_1}{r-2}] + \frac{4}{r-2}\left(\frac{E[Y_H]}{2^H} - \frac{E[Y_1]}{2}\right) \quad \text{(since } F_1 \leq \frac{Y_1}{r} \leq \frac{Y_1}{(r-2)}\text{)}$$

$$= \frac{4}{r-2}\left(\frac{E[Y_1]}{2} + \frac{E[Y_H]}{2^H} - \frac{E[Y_1]}{2}\right)$$

$$\leq \frac{4}{r-2}\frac{E[Y_H]}{2^H} \quad \text{(note that } Y_H \text{ is the number of times } \pi_{opt} \text{ crosses the edges of } G^H\text{)}$$

$$= \frac{4}{r-2}\frac{2^{H+1}OPT}{2^H} \quad \text{(due to (1))}$$

error type (ii): patching π_{opt} at every level from bottom-to-top

F_i: number of patching operations performed at i^{th} level; Y_i: number of times the current tour crosses the edges of G^i just before considering G^i for patching; n_i: number of times the current tour crosses the edges of G^i after patching

expected increase in length of π_{opt} due to patchings at every level

$$= E[\sum_{i=1}^{H} \frac{4F_i}{2^i}] \quad \text{(due to (0))}$$

$$\leq E[2F_1 + \sum_{i=2}^{H} \frac{4F_i}{2^i}]$$

$$\leq E[2F_1] + \frac{4}{r-2} \sum_{i=2}^{H} \frac{E[Y_i]-2E[Y_{i-1}]}{2^i}$$

(noting that due to (2), $E[Y_{i-1}] = E[E[Y_{i-1}|n_i]] = E[\frac{n_i}{2}]$, and

$E[Y_i] = E[\frac{n_i}{2}] \leq E[\frac{Y_i-(r-2)F_i}{2}] = \frac{1}{2}E[Y_i] - \frac{k-2}{2}E[F_i]$)

$$\leq E[2\frac{Y_1}{r-2}] + \frac{4}{r-2}(\frac{E[Y_H]}{2^H} - \frac{E[Y_1]}{2}) \quad \text{(since } F_1 \leq \frac{Y_1}{r} \leq \frac{Y_1}{(r-2)}\text{)}$$

$$= \frac{4}{r-2}(\frac{E[Y_1]}{2} + \frac{E[Y_H]}{2^H} - \frac{E[Y_1]}{2})$$

$$\leq \frac{4}{r-2}\frac{E[Y_H]}{2^H} \quad \text{(note that } Y_H \text{ is the number of times } \pi_{opt} \text{ crosses the edges of } G^H\text{)}$$

$$= \frac{4}{r-2}\frac{2^{H+1}OPT}{2^H} \quad \text{(due to (1))}$$

$$= \frac{8}{r-2}OPT$$

error type (iii): expected error due to snapping tour to portals

- as noted, after applying patching lemma to grid squares at all levels from bottom-to-top, the tour intersects with any edge of any grid square at most twice

[7] for $i \geq 1$, tour π_i (esp. π_1) has at least as many crossings with G_i as π_{opt} has with G_i

[8] if a grid edge has portals from several levels, the snapped tour used only the portals on this edge that belong to the highest level

error type (iii): expected error due to snapping tour to portals

- as noted, after applying patching lemma to grid squares at all levels from bottom-to-top, the tour intersects with any edge of any grid square at most twice

- as the distance between any two successive portals along any edge of G^i is $\frac{1/2^i}{m+1}$, a single snapping operation on an edge of G^i introduces an error $\leq 2(\frac{1/2^i}{2(m+1)}) = \frac{1}{2^i(m+1)}$

[7] for $i \geq 1$, tour π_i (esp. π_1) has at least as many crossings with G_i as π_{opt} has with G_i

[8] if a grid edge has portals from several levels, the snapped tour used only the portals on this edge that belong to the highest level

error type (iii): expected error due to snapping tour to portals

- as noted, after applying patching lemma to grid squares at all levels from bottom-to-top, the tour intersects with any edge of any grid square at most twice

- as the distance between any two successive portals along any edge of G^i is $\frac{1/2^i}{m+1}$, a single snapping operation on an edge of G^i introduces an error $\leq 2(\frac{1/2^i}{2(m+1)}) = \frac{1}{2^i(m+1)}$

- let Z_i be the number of intersections of π_{opt} with G^i for $i \geq 1$ [7]

  due to (1), $E[Z_i] \leq 2^{i+1} OPT$

[7] for $i \geq 1$, tour π_i (esp. π_1) has at least as many crossings with G_i as π_{opt} has with G_i

[8] if a grid edge has portals from several levels, the snapped tour used only the portals on this edge that belong to the highest level

error type (iii): expected error due to snapping tour to portals

- as noted, after applying patching lemma to grid squares at all levels from bottom-to-top, the tour intersects with any edge of any grid square at most twice

- as the distance between any two successive portals along any edge of G^i is $\frac{1/2^i}{m+1}$, a single snapping operation on an edge of G^i introduces an error $\leq 2(\frac{1/2^i}{2(m+1)}) = \frac{1}{2^i(m+1)}$

- let Z_i be the number of intersections of π_{opt} with G^i for $i \geq 1$ [7]

  due to (1), $E[Z_i] \leq 2^{i+1} OPT$

- expected increase in length due to snapping π_1 to the portals of the quadtree (summed over all the levels)[8] $\leq E[\sum_{i=1}^{H} \frac{Z_i}{2^i(m+1)}] \leq \frac{2H}{m+1} OPT$

---

[7] for $i \geq 1$, tour π_i (esp. π_1) has at least as many crossings with G_i as π_{opt} has with G_i

[8] if a grid edge has portals from several levels, the snapped tour used only the portals on this edge that belong to the highest level

overall apprx factor

- $(1+\frac{\epsilon}{2})(1+\frac{8}{r-2})(1+\frac{2H}{m+1})OPT$

  $\leq (1+\frac{\epsilon}{2})(1+\frac{\epsilon}{10})^2 OPT$ (by choosing $r=\frac{90}{\epsilon}=O(\frac{1}{\epsilon})$ and $m \geq \frac{20H}{\epsilon}=O(\frac{\lg n}{\epsilon})$)

  $\leq (1+\epsilon)OPT$

overall apprx factor

- $(1+\frac{\epsilon}{2})(1+\frac{8}{r-2})(1+\frac{2H}{m+1})OPT$

  $\leq (1+\frac{\epsilon}{2})(1+\frac{\epsilon}{10})^2 OPT$ (by choosing $r=\frac{90}{\epsilon}=O(\frac{1}{\epsilon})$ and $m \geq \frac{20H}{\epsilon}=O(\frac{\lg n}{\epsilon})$)

  $\leq (1+\epsilon)OPT$

hence, the apprx factor in expectation is $(1+\epsilon)$ and the time complexity is $n(\frac{\lg n}{\epsilon})^{O(1/\epsilon)}$ i.e., randomized PTAS

take-home lessons

- PTAS in expectation
- rounding input points
- Steiner points (portals) along grid edges
- dynamic programming
- locally patching a path
- Hochbaum-Mass shifting grid technique
- applicable in various contexts: minimum Steiner tree, k-TSP, k-MST, Euclidean min-cost perfect matching

hence, the Godel award for the result!

other important results

- $O(n^{d+1} polylog)$ time PTAS — Arora '97
- $O(n \lg n)$ time PTAS — Rao and Smith '98
- NP-hard to get $(1+\epsilon)$-apprx in $\mathbb{R}^{O(\lg n)}$, for some $\epsilon > 0$ — Trevisan '97

references

- S. Har-Peled. Geometric Approximation Algorithms. American Mathematical Society, 2011.
- D. P. Williamson and D. B. Shmoys. The Design of Approximation Algorithms. Cambridge University Press, 2011.
- S. Arora. Polynomial time approximation schemes for Euclidean TSP and other geometric problems. Journal of ACM, 1998.

references

- S. Har-Peled. Geometric Approximation Algorithms. American Mathematical Society, 2011.
- D. P. Williamson and D. B. Shmoys. The Design of Approximation Algorithms. Cambridge University Press, 2011.
- S. Arora. Polynomial time approximation schemes for Euclidean TSP and other geometric problems. Journal of ACM, 1998.

Parallel/further significant work -

- J. S. B. Mitchell. Guillotine subdivisions approximate polygonal subdivisions: A simple new method for the geometric TSP, k-MST, and related problems. SIAM Journal on Computing, 1999.
- S. B. Rao and W. D. Smith. Approximating geometric graphs via ”spanners” and ”banyans”. Symposium on Theory of Computing, 1998.